Applied Soft Computing 16 (2014) 20–33

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Differential evolution based 3-D guidance law for a realistic interceptor model

T. Raghunathan [a,*,1], D. Ghose [b]

[a] School of Electrical Engineering, VIT University, Vellore, India
[b] Guidance, Control and Decision Systems Laboratory, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India

ARTICLE INFO

Article history:
Received 20 December 2012
Received in revised form
12 September 2013
Accepted 29 November 2013
Available online 8 December 2013

Keywords:
Missile guidance
Optimal control
Optimal trajectory
Genetic algorithm
Online control
Real-time

ABSTRACT

This paper presents a novel, soft computing based solution to a complex optimal control or dynamic optimization problem that requires the solution to be available in real-time. The complexities in this problem of optimal guidance of interceptors launched with high initial heading errors include the more involved physics of a three dimensional missile–target engagement, and those posed by the assumption of a realistic dynamic model such as time-varying missile speed, thrust, drag and mass, besides gravity, and upper bound on the lateral acceleration. The classic, pure proportional navigation law is augmented with a polynomial function of the heading error, and the values of the coefficients of the polynomial are determined using differential evolution (DE). The performance of the proposed DE enhanced guidance law is compared against the existing conventional laws in the literature, on the criteria of time and energy optimality, peak lateral acceleration demanded, terminal speed and robustness to unanticipated target maneuvers, to illustrate the superiority of the proposed law.

1. Introduction

Interception of an aerial target by missiles or interceptors is an important practical problem. However, this is far from amenable to any easy solutions due to the inherent complexities of the problem. Realistic models of missile dynamics must include the effect of time-varying missile speed, thrust, drag and mass, besides gravity. The drag in turn is a function of the missile speed, atmospheric density, and lateral acceleration (latax). Another important practical constraint is the upper bound on the latax. Two simplifications resorted to in the literature are (i) consider only the kinematics, and (ii) separate the three dimensional (3-D) engagement into two planar (or two dimensional, or 2-D) kinematic models in the horizontal and vertical plane. Enabled by such gross simplifications, analytical or closed form guidance laws that can be easily implemented at every time instant, and with minimum computation, are formulated or derived, as in [56,38]. Given the much higher complexity of analysis and design in the 3-D space, guidance papers pertaining to 3-D engagements are far fewer in number, as compared to those for 2-D engagements, in spite of the fact that the first paper on 3-D guidance was published as early as in the 1950s itself [1].

Even for the simpler planar kinematic model, interception is not an easy task if the initial engagement geometry has high heading errors. Determination of the guidance law in a high heading error engagement, subject to the kinematic equation alone, has been the subject of many papers and techniques, and some of these techniques are based on methods of nonlinear control design. Some examples are: feedback linearization [2], relative heading error angle concept [22], state-dependent Riccati equation technique [9], extended proportional navigation [54], variable structure control [18,17], and, a combination of LOS rate and heading error [49]. Of these, only Cloutier and Stansbery [9] and Taur [49] deal with 3-D kinematic engagements; the rest consider only 2-D kinematic models.

The scenario becomes even more complicated if one imposes the additional requirement of optimality. Analytical optimal guidance laws derived from kinematic models require time-to-go parameter for their implementation, due to the free final-time nature of the problem. Consequently, accurate estimation of time-to-go itself has been another continuing topic of research, as in Ryoo et al. [35,36] and Tahk et al. [48].

The nonlinear optimal control problem applicable to high heading error initial geometries is a two point boundary value problem that is amenable to only numerical solutions and demand considerable computational effort, thereby not being implementable in

[*] Corresponding author. Tel.: +91 9489969616; fax: +91 416 2243092.
E-mail addresses: raghunathan.t@vit.ac.in, nathraghu@yahoo.com (T. Raghunathan), dghose@aero.iisc.ernet.in (D. Ghose).

[1] Formerly at: Guidance, Control and Decision Systems Laboratory, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India.

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.11.017

real-time. Further, the performance of these numerical solutions deteriorates for even small deviations between the model assumed for design and the actual plant on which they are applied, due to the inherently open loop nature of these solutions. This makes the case for finding closed loop solutions or guidance laws, that are implementable in real-time, using a more realistic dynamic model, without resorting to the unrealistic simplifying assumptions for the sake of mathematical tractability.

The present paper formulates and studies the performance of an online-implementable optimal guidance law based on a realistic model of engagement that explicitly includes all the constraints mentioned in the beginning of Section 1, in 3-D physical space, for high initial heading errors. Except for treating the missile and target as point masses, no other approximations are made. The large number of inequality constraints in the formulation itself makes it very difficult to obtain an analytical or closed form solution, making numerical solution the only option. Any optimal control problem, linear or nonlinear, with any number of compatible equality and inequality constraints on state and controls can be solved by dynamic programming [28]. Another method of solution of the nonlinear optimal control problem is by converting it to a nonlinear programming (NLP) problem and solving the NLP problem, as in Cruz et al. [10]. A method that has been applied in optimal missile guidance, for a realistic model – though only 2-D – is Imado, Kuroda and Miwa [16]. However, as has been acknowledged sometimes by the authors themselves [16], all these are too computationally intensive for implementation in real-time.

In this paper, the optimal guidance problem is posed as a low dimensional polynomial optimization problem, and a numerical solution to this problem, computable in real-time, is obtained. The basic pure proportional navigation (PPN) law is augmented with a polynomial function of the heading error, and optimization is achieved by using DE to find the values of the coefficients of the polynomial. The proposed law is then compared with the all-aspect proportional navigation (AAPN) [39], and PPN laws on the following criteria: energy and time optimality, the peak latax demanded, terminal speed, and robustness to off-nominal conditions in the form of unanticipated target maneuvers. The AAPN law of Sim et al. [39] used the same structure as in the guidance law proposed in the present paper, but the solution proposed was applicable for the given initial conditions only, and needed to be recomputed for every different set of initial condition, as pointed out by the authors themselves. It also assumed the kinematic model of engagement. Moreover, the solution methodology was time consuming and hence unsuitable for online implementation. The law proposed in the present paper is applicable to any initial conditions, for a realistic dynamic model of engagement, and is also online implementable.

In the previous work of the authors [31], a DE based optimal guidance law was proposed for a two dimensional (2-D) kinematic model of missile–target engagement. It was a proof of concept that, for optimal missile guidance, evolutionary algorithms like the DE could serve as an alternative to the optimal control based methods, thereby circumventing the various difficulties posed by these methods. The work described in the present paper is much more elaborate, since it uses a 3-D realistic engagement model that includes missile dynamics and aerodynamic effects.

Compared to the previous work of the authors [31] which considered engagements with a simple, planar kinematic model, the present work assumes a 3-D realistic model that introduces the following complications: First, the state variables are subject to numerous interdependent equality and inequality constraints. Since these are indirect functions of the altitude, they depend on the vertical position of the missile in the inertial frame of reference. Secondly, the latax is applied by the missile in its own body frame of reference, and the LOS rates are measured and heading error angles are readily computed in the LOS frame of reference. This entails conversion from one frame of reference to another. Thirdly, unlike in the kinematic model in which the missile speed is constant, missile speed also enters the state as a variable, and is characterized by sharp discontinuities as the missile thrust switches from boost to sustain, and sustain to coast phases.

Fourthly, a very serious practical difficulty is that, with the dynamics also taken into account in the realistic model, the critical constraining factor is the thrust that is available only for a limited time period. The missile speed that is highly dependent on the thrust begins to fall off from the end of the boost phase, especially if the missile is still applying a large latax. Any practical interception is considered to be successful only if the missile speed is high enough just before interception, so that the missile is in a position to engage the target in the end game scenario. Hence, for a successful interception, the missile thrust should still be in the sustain phase, or at least not be too long into the coast phase.

The guidance law designed must take into account all these factors, and hence moving from a 2-D kinematic model to a 3-D realistic model is far from a mere extension of the guidance law designed for the 2-D kinematic model. The fifth and final complication that is of crucial importance is, the solution or the guidance law that involves the four complications described above, has to be available or computable in real time, to be of any practical relevance. The main contribution of this paper is in providing a real-time solution that is online implementable to this challenging problem, using differential evolution as the enabling method. Since the problem is a real world one, there is no generic, elegant solution; problem specific knowledge has to be incorporated, and real world complications have to be addressed, as described in the paper.

The paper is organized as follows: Section 2 presents the mathematical preliminaries, the 3-D PPN, and all-aspect proportional navigation (AAPN) laws, and the development of the 3-D counterpart of the 2-D differential evolution tuned all-aspect proportional navigation (DEPN-2D) law. Section 3 explains the DE algorithm used in the paper, Section 4 describes the design of the proposed 3-D guidance law using DE, Section 5 presents the simulation results and comparison of the proposed law with the other guidance laws, and is followed by concluding remarks in Section 6.

2. Vehicle models and guidance laws in 3-D engagement scenario

Using a Cartesian co-ordinate representation in an inertial frame of reference, the three dimensional (3-D) point-mass equations of motion for the missile and the target are modelled as follows:

Missile:

$$\begin{bmatrix} \dot{x_m} \\ \dot{y_m} \\ \dot{z_m} \\ \dot{\phi_m} \\ \dot{\gamma_m} \\ \dot{V_m} \end{bmatrix} = \begin{bmatrix} V_m \cos\gamma_m \cos\phi_m \\ V_m \cos\gamma_m \sin\phi_m \\ V_m \sin\gamma_m \\ a_{ym}/(V_m \cos\gamma_m) \\ (a_{pm} - g\cos\gamma_m)/V_m \\ (\Gamma - D)/m - g\sin\gamma_m \end{bmatrix} \tag{1}$$

$$a_{ym}^2 + a_{pm}^2 \le \overline{a}_m^2 \tag{2}$$

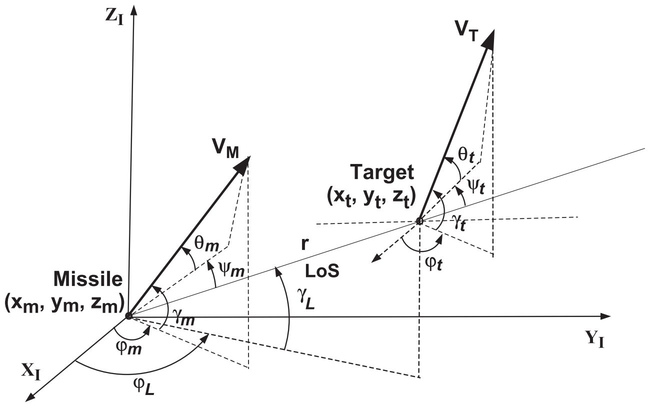


Fig. 1. 3-D missile–target engagement geometry.

Target:

$$\begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \\ \dot{\phi}_t \\ \dot{\gamma}_t \end{bmatrix} = \begin{bmatrix} V_t \cos\gamma_t \cos\phi_t \\ V_t \cos\gamma_t \sin\phi_t \\ V_t \sin\gamma_t \\ a_{yt}/(V_t \cos\gamma_t) \\ a_{pt}/V_t \end{bmatrix} \tag{3}$$

$\Delta x = x_t - x_m, \quad \Delta y = y_t - y_m, \quad \Delta z = z_t - z_m r = \sqrt{(\Delta x^2 + \Delta y^2 + \Delta z)}^{2})\gamma_L = \tan^{-1}(\Delta z/\sqrt{(\Delta x^2 + \Delta y^2)}), \quad \phi_L = \tan^{-1}(\Delta y/\Delta x)$ where V_m is the missile speed, Γ the time-varying thrust, D the drag, m the mass of the missile, and g the acceleration due to gravity. a_{ym}, a_{pm}, a_{yt} and a_{pt} are the yaw and pitch latax's of the missile and target, respectively, and $\bar{a}_m$ represents the maximum permissible latax of the missile. The angles ϕ_m, γ_m, ϕ_t, γ_t are Euler angles made by the missile and the target (Fig. 1). The target speed V_t is assumed to be constant.

2.1. The 3-D PPN law

A guidance law that has been proved to guarantee interception of the target, whether maneuvering or non-maneuvering, – for the kinematic model – is the pure proportional navigation (PPN) law [13,14]. The 3-D PPN law is given by [41]

$$\mathbf{a}_M = N\Omega_L \times \mathbf{V}_M \tag{4}$$

where $\mathbf{a}_M$ is the missile latax vector, N the navigation constant, Ω_L the angular velocity of the LOS, and $\mathbf{V}_M$ the missile velocity vector.

In the LOS frame of reference,

$$\Omega_L = \dot{\lambda}_x i_L + \dot{\lambda}_y j_L + \dot{\lambda}_z k_L \tag{5}$$

$$\mathbf{V}_M^L = V_{mx}^L i_L + V_{my}^L j_L + V_{mz}^L k_L \tag{6}$$

where $\dot{\lambda}_x$, $\dot{\lambda}_y$, $\dot{\lambda}_z$ are the X_L, Y_L, Z_L components, respectively, of the LOS angular velocity vector, i_L, j_L, k_L are the unit vectors along, respectively, to X_L, Y_L, Z_L axes, $\mathbf{V}_M^L$ is the missile velocity vector $\mathbf{V}_M$ in LOS frame of reference, and V_{mx}^L, V_{my}^L, V_{mz}^L are the missile velocity vector components, respectively, along X_L, Y_L, Z_L axes, in the LOS frame of reference, and given by $V_{mx}^L = V_m \cos\theta_m \cos\psi_m$, $V_{my}^L = V_m \cos\theta_m \sin\psi_m$, $V_{mz}^L = V_m \sin\theta_m$, with the angles θ_m and ψ_m defined with respect to the LOS (Fig. 1).

However, $\dot{\lambda}_x$ cannot be measured by the conventional on-board seeker [41]. Hence, using only the last two terms of (5), and (6) in (4),

$$\mathbf{a}_M^L = N\{(V_{mz}^L \dot{\lambda}_y - V_{my}^L \dot{\lambda}_z) i_L + V_{mx}^L \dot{\lambda}_z j_L - V_{mx}^L \dot{\lambda}_y k_L\} \tag{7}$$

where $\mathbf{a}_M^L$ is the missile latax vector in the LOS frame of reference.

For practical implementation, this latax needs to be converted to the missile body frame of reference by using

$$\mathbf{a}_M^m = T_{Lm} \mathbf{a}_M^L \tag{8}$$

where the co-ordinate transformation matrix T_{Lm} that achieves the transformation from the LOS frame of reference to the missile body frame of reference is given by

$$T_{Lm} = \begin{bmatrix} \cos\psi_m \cos\theta_m & \sin\psi_m \cos\theta_m & \sin\theta_m \\ -\sin\psi_m & \cos\psi_m & 0 \\ -\cos\psi_m \sin\theta_m & -\sin\psi_m \sin\theta_m & \cos\theta_m \end{bmatrix} \tag{9}$$

The latax $\mathbf{a}_M^m$ in (8) has two components in the yaw and pitch plane of the missile:

$$\mathbf{a}_M^m = a_{ym} j_m + a_{pm} k_m \tag{10}$$

$$a_{ym} = -NV_m \dot{\lambda}_y \sin\theta_m \sin\psi_m + NV_m \dot{\lambda}_z \cos\theta_m \tag{11}$$

$$a_{pm} = -NV_m \dot{\lambda}_y \cos\psi_m \tag{12}$$

$$\dot{\lambda}_y = \frac{V_m \sin\theta_m - V_t \sin\theta_t}{r} \tag{13}$$

$$\dot{\lambda}_z = \frac{V_t \cos\theta_t \sin\psi_t - V_m \cos\theta_m \sin\psi_m}{r} \tag{14}$$

where a_{ym} and a_{pm} are along the Y_m and Z_m axes of the missile frame of reference, and the angles θ_t and ψ_t are defined with respect to the LOS frame of reference (Fig. 1).

2.2. The 3-D differential evolution tuned all-aspect proportional navigation law

The PPN law that meets the essential requirement of interceptability can be modified to meet the desirable requirement of optimality by augmenting it with an additional term for this purpose, leading to the AAPN law. The AAPN guidance law for 2-D engagement is given by [39]

$$a_{m,\,2D} = NV_m \dot{\lambda} + f(h, \quad V_{ratio}) \left(\frac{V_m^2}{r} \right) \tag{15}$$

where f is a polynomial function of the heading error angle h, and $V_{ratio} = V_t/V_m$. Since f is far less sensitive to V_{ratio} as compared to h [39], f reduces to

$$f = ah + bh^2 + ch^3 \tag{16}$$

With this, interception-guaranteed optimality was sought to be ensured, for the assumed kinematic model. However, the issue of determining a, b, and c was not a simple task that could be achieved in real-time. It was calculated offline in Sim et al. [39], using a protracted three step procedure. A simple one step procedure that was implementable online was proposed by Raghunathan and Ghose [31], and named the DE-AAPN law, and referred to as the DEPN-2D law, to distinguish it from the 3D counterpart proposed in this paper. The DEPN-2D law uses DE to find out the values of the coefficients a, b and c.

As with the latax vector in the 3-D PPN law given by (10), the 3-D equivalent of the DEPN-2D law is given by two components

$$\mathbf{A}_M = A_{ym} j_m + A_{pm} k_m \tag{17}$$

The assumption made here is the same as in (10): the missile acceleration comprises only lateral acceleration (latax), and contains no component along the longitudinal direction.

The pitch and the yaw latax components A_{ym} and A_{pm} in (17) are given by

$$A_{pm} = a_{pm} + f_p(h_p)\left(\frac{V_m^2}{r}\right) \tag{18}$$

$$A_{ym} = a_{ym} + f_y(h_y)\left(\frac{V_m^2}{r}\right) \tag{19}$$

$$f_p = a_p h_p + b_p h_p^2 + c_p h_p^3 \tag{20}$$

$$f_y = a_y h_y + b_y h_y^2 + c_y h_y^3 \tag{21}$$

where h_p and h_y are the heading errors in the pitch and yaw planes in the LOS frame of reference. This is in contrast to the 2-D case, in which only a single heading error exists (16). As in the 2-D case, the DEPN-3D law involves finding out the values of the coefficients a_p, b_p, c_p, a_y, b_y and c_y using DE.

The angles $\theta_m, \psi_m, \theta_t, \psi_t, h_y$ and h_p in (11)–(14) and (18)–(21) are in the LOS frame of reference, whereas the states of the missile and the target being subject to altitude dependent practical constraints like density and temperature, have to be obtained by integration of (1) and (3) in the inertial frame of reference. This needs co-ordinate transformation from inertial to the LOS frame to be done at every time instant of the simulation.

$$\mathbf{V}_M^L = T_{IL}\mathbf{V}_M^I \tag{22}$$

$$\mathbf{V}_T^L = T_{IL}\mathbf{V}_T^I \tag{23}$$

where $\mathbf{V}_M^I$ and $\mathbf{V}_T^I$ are the missile and target velocity vectors $\mathbf{V}_M$ and $\mathbf{V}_T$ in the inertial frame of reference, given by

$$\mathbf{V}_M^I = V_{mx}^I i_I + V_{my}^I j_I + V_{mz}^I k_I \tag{24}$$

$$\mathbf{V}_T^I = V_{tx}^I i_I + V_{ty}^I j_I + V_{tz}^I k_I \tag{25}$$

where i_I, j_I, k_I are the unit vectors along X_I, Y_I, Z_I axes, respectively, and V_{mx}^I, V_{my}^I, V_{mz}^I are the missile velocity vector components, respectively, along X_I, Y_I, Z_I axes, in the inertial frame of reference, and given by $V_{mx}^I = V_m\cos\gamma_m\cos\phi_m$, $V_{my}^I = V_m\cos\gamma_m\sin\phi_m$, $V_{mz}^I = V_m\sin\gamma_m$. Similarly, $V_{tx}^I = V_t\cos\gamma_t\cos\phi_t$, $V_{ty}^I = V_t\cos\gamma_t\sin\phi_t$, $V_{tz}^I = V_t\sin\gamma_t$.

The co-ordinate transformation matrix that achieves the transformation from inertial to the LOS frame of reference T_{IL} is given by

$$T_{IL} = \begin{bmatrix} \cos\phi_L\cos\gamma_L & \sin\phi_L\cos\gamma_L & \sin\gamma_L \\ -\sin\phi_L & \cos\phi_L & 0 \\ -\cos\phi_L\sin\gamma_L & -\sin\phi_L\sin\gamma_L & \cos\gamma_L \end{bmatrix} \tag{26}$$

and

$$T_{LI} = T'_{IL} \tag{27}$$

$$\mathbf{V}_T^L = V_{tx}^L i_L + V_{ty}^L j_L + V_{tz}^L k_L \tag{28}$$

where $V_{tx}^L = V_t\cos\theta_t\cos\psi_t$, $V_{ty}^L = V_t\cos\theta_t\sin\psi_t$, and $V_{tz}^L = V_t\sin\theta_t$. The angles h_p, h_y in (18) and (19) are given by (Fig. 2(a) and (b))

$$h_p(t) = \theta_m(t) - L_p(t) \tag{29}$$

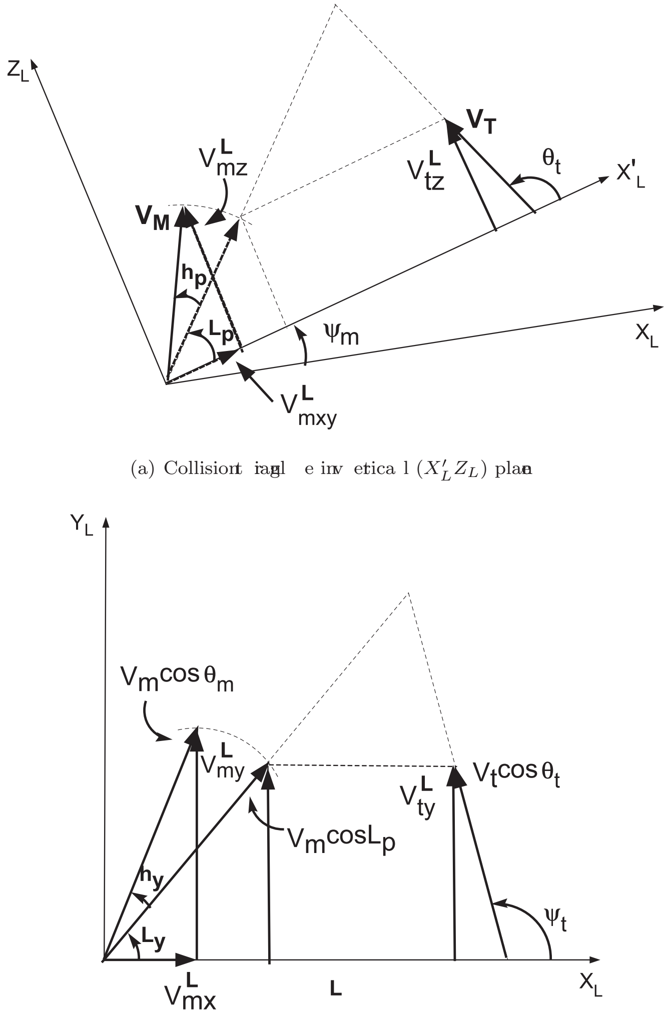
$$h_y(t) = \psi_m(t) - L_y(t) \tag{30}$$

$$\theta_m(t) = \tan^{-1}(V_{mz}^L(t)V_{mxy}^L(t)) \tag{31}$$

$$V_{mxy}^L(t) = \sqrt{\{V_{mx}^L(t)\}^2 + \{V_{my}^L(t)\}^2} \tag{32}$$

$$L_p(t) = \sin^{-1}(V_{tz}^L(t)V_m(t)) \tag{33}$$

$$\psi_m(t) = \tan^{-1}(V_{my}^L(t)/V_{mx}^L(t)) \tag{34}$$



(a) Collision triangle in vertical ($X'_L Z_L$) plane

(b) Collision triangle in horizontal ($X_L Y_L$) plane

Fig. 2. Collision triangles in 3-D space in LOS frame of reference.

$$L_y(t) = \sin^{-1}\left\{\frac{V_{ty}^L(t)}{V_m(t)\cos L_p(t)}\right\} \tag{35}$$

3. Evolutionary algorithms and the differential evolution algorithm

A later addition to the EA family, the differential evolution (DE) was proposed by Price and Storn [45,46,42,30,47,29,43]. A popular member of the evolutionary algorithm (EA) family, the DE owes its popularity to its simplicity of implementation, and performance in terms of reproducibility of results, and relative insensitivity to change in its tuning parameters, that have been proven at several congress on evolutionary computation (CEC) competitions [12]. An empirical study of some basic versions of the particle swarm optimization (PSO), genetic algorithm (GA) and DE algorithms, found DE to be the best performer on a test suite of static optimization benchmark problems, and regarded the DE as an excellent first choice when faced with a new (static) optimization problem [51]. Another comparison of the evolutionary computation (EC) algorithms of the GA, DE, evolutionary programming (EP), evolutionary strategy (ES), ant colony optimization (ACO), PSO, and the other algorithms of Tabu search, simulated annealing, and hybrid approach (HA) is found in Kannan et al. [19]. These methods have been applied to the real-world, highly constrained, nonlinear, discrete optimization problem of generation expansion planning, and it was found that, excepting the HA, the DE outperformed all the others in terms of success rate (SR), defined as the number of times the optimal solution was achieved to the total number of test runs. It was opined that this was due to the nature of mutation employed by DE. This provides DE with one of its main assets, 'contour matching' [44], which refers to the self-adaptive property of the vector population to automatically explore the most promising regions of the solution space, once they are detected. A good survey of DE and the state-of-the-art including its major variants, application to difficult optimization problems and engineering applications is covered in Das and Suganthan [12] and Chakraborthy [4]. A generic explanation of how the DE algorithm works and the pseudocode can be found in Raghunathan and Ghose [31].

3.1. EC, DE and the optimal control problem

Given the inherent advantages of EC over other gradient based optimization methods, the rich potential of EC methods for solving optimal control problems were recognized in the 1990s itself [26], [25] and [21]. As these were simply demonstrations of proof of concept, most of the problems solved in these were simple optimal control problems for which analytic solutions existed. This was followed by problems of increasing sophistication like in Seywald et al. [37] and Sim et al. [39], in which the GA was used to provide good initial guess solution for the methods used therein. Other than those of the authors of the present paper, probably the only applications of DE to optimal control problems are Chiou and Wang [7] and Cruz et al. [10], in which DE is applied to get offline solutions to multimodal problems for which the solutions are known beforehand. In Sim et al. [39], the optimal control problem solved was an optimal missile guidance problem for a kinematic model. Here too, the solution was obtained offline, based on an existing solution obtained by a different method. To the best of our knowledge, the first online implementable solution to the optimal missile guidance problem obtained by an EA was in Raghunathan and Ghose [31]. In this work, the problem solved offline in Sim et al. [39] was solved online in real-time using DE, since it was felt that, given the inherent simplicity of the DE, it would be the most suitable EA for online implementation. The present work too uses DE for the same reason, but for the more complex, realistic, 3-D missile–target engagement model.

4. Design of DEPN-3D law using DE

Several empirical studies on the performance of the different DE variants have appeared in the literature over the years. Different DE variants, both classic and more sophisticated, are applied to a suite of benchmark functions, and the results summarized. These functions varied in parameter dependence, dimensionality, modality and noise. This is the approach followed in, for instance, Mezura-Montes et al. [24], Brest et al. [3], Das et al. [11] and Zhang and Sanderson [57]. But these are all static optimization problems, and applications of DE to dynamic optimization or optimal control problems itself are very few (cited in Section 3.1), and no such study has been performed for optimal control problems, to the best of the authors' knowledge.

Obviously, no optimization method can perform equally well on all optimization problems, as explained by the No Free Lunch (NFL) Theorem [52]. Consequently, the method that works best for any particular problem has to be found by trial and error by trying out all the methods. Applying this to the DE variants, the variant that works best for the optimal control problem or optimal missile guidance has to be found out empirically, by trying out all of them. This is outside the scope of the present work, and may be considered for some future work. The scope of the present paper is limited to application of classic DE to optimal missile guidance using a realistic 3-D model.

To apply DE to the missile guidance problem, the values of the DE parameters are first determined. Several claims and counterclaims are found in the DE literature concerning the rules for tuning of the DE parameters, but these are not backed by experimental justifications [12]. Hence the approach adopted here is to begin with a few guidelines available in DE literature, and fine-tune the algorithm by executing a few initial sample runs.

4.1. Some issues in application of DE to missile guidance

4.1.1. Choice of tuning parameters N_P, C_r and F

A detailed study of the best values of F and C_r would call for a theoretical analysis as well as empirical study, as in Zaharie [55]. Since such a study is out of the scope of this paper, a more practical and immediate way is adopted. Storn [44] gives a 'rule of thumb' guideline of the approximate range from which to begin while trying to solve a problem by DE: mutation (or differentiation) constant $F \in [0.5, 1.0]$, crossover ratio/constant $C_r \in [0.8, 1.0]$, population size $N_P = 10D$, where D is the number of unknown variables, or the dimensionality of the problem. There also exist the default values from which to begin [43]: $C_r = 0.9$, and $F = 0.8$. Given the unknown solution surface topology, this default, high value of F that leans towards exploration, rather than exploitation, is retained. The high value of C_r also implies that the trial vector is more of a mutant than parent, but the problem is found to work well for this default value as well. As it was observed that the problem was not sensitive to reasonably small variations in the values of these DE parameters, the default values were used, without any further study of this aspect of the algorithm.

The suggested value of $N_P = 10D$ necessitates $10D$ cost function evaluations in each generation. Since an indispensable requirement of the missile guidance problem is online implementability, and cost function evaluation is found to be the most computationally intensive of all steps in the whole procedure, the population size N_P is sought to be reduced as the first step. The mutation and crossover operators of DE impose an absolute lower limit ($N_P \nless 4$), but since this is too low for a multi-point directed random search algorithm like DE, $N_P = 4D$, arrived at by trial and error, and traded off against the maximum number of generations G_{max}, as in Raghunathan and Ghose [31].

4.1.2. DE variant used

The classic DE variant used is DE/best/1/bin, as against DE/rand/1/bin, another choice that works equally well for the problem in this paper. Here the 'best' or 'rand' in the above notation [44] refers to the base vector that is perturbed in mutation. 'best' means the fittest in the generation, and 'rand' means that the vector is chosen randomly. The '1' refers to the number of difference vectors used, and 'bin' refers to the binomial crossover scheme. More details and comparison of these variants can be found in Storn [44] and Price [29].

4.1.3. Bounds on the solution space

The use of [0,1] as the bounds for $|a|$, $|b|$ and $|c|$ in the AAPN law of Sim et al. [39] assumes that the scaling factor V_m^2/r in (15) scales the values of a, b and c to within ±1. Through simulations, it was found that better and faster solutions are obtained by enlarging the solution space to beyond ±1, probably because the above assumption does not always hold, especially when large heading errors are involved. On the other hand, too large a range would mean needlessly enlarging the search space, something that is not in the interest of online implementation. By trial and error, the bounds for the solution space is fixed as $[-5, 5]$.

4.1.4. The fitness function

The DEPN-3D law solves the following optimal control problem

$$\min_{a,b,c} \; J = \frac{1}{2}\int_{t_0}^{t_f} (A_m^2 + W_t)dt \quad \text{subject to} \quad (1)\text{–}(3) \tag{36}$$

where $A_m = \|\mathbf{A}_M\| = \sqrt{(A_{pm}^2 + A_{ym}^2)}$, with A_{pm} and A_{ym} replacing a_{pm} and a_{ym}, respectively, in (1) and (2). In EA terminology, the J used in (36) is the fitness function used for evaluation of the trial solutions.

4.1.5. Latax saturation

The latax saturation that is represented by the inequality constraint (2) has to be implemented on two separately computed components (18) and (19). A practical way of implementing it that is used in this paper is

$$A_{pm}(A_{ym}) = \begin{cases} A_{pm}^c(A_{ym}^c) & \text{if } \sqrt{(A_{pm}^c)^2 + (A_{ym}^c)^2} \le \bar{a}_m \\ \dfrac{\bar{a}_m}{\sqrt{(A_{pm}^c)^2 + (A_{ym}^c)^2}} A_{pm}^c(A_{ym}^c) & \text{otherwise} \end{cases} \tag{37}$$

where A_{pm}^c and A_{ym}^c are the pitch and yaw latax components computed by (18) and (19).

4.2. The DE solution process

4.2.1. Initialization

The initial zeroth generation population $P_{G=0}$ of N_P trial solutions or individuals are sextuples of $(a_p, b_p, c_p, a_y, b_y, c_y)$, generated randomly within the range ±5.

$$P_{G=0} = \{\bar{x}_{1,0}, \bar{x}_{2,0}, \ldots \bar{x}_{i,0}, \ldots, \bar{x}_{N_P,0}\} \tag{38}$$

where

$$\begin{aligned} \bar{x}_{i,0} &= \{x_{1,i,0}, x_{2,i,0}, x_{3,i,0}, x_{4,i,0}, x_{5,i,0}, x_{6,i,0}\} \\ &= \{a_{p,i,0}, b_{p,i,0}, c_{p,i,0}, a_{y,i,0}, b_{y,i,0}, c_{y,i,0}\} \\ p_{i,G=0} &= -5 + rand_j[0,1] \times (+5 - (-5)) \\ p &= a_p, b_p, c_p, a_y, b_y \;\; \text{or} \;\; c_y, i = 1, 2, \ldots, N_P \end{aligned}$$

4.2.2. Mutation and crossover

The mutant vector $v_{i,G+1}$ is produced with $F = 0.8$:

$$v_{i,G+1} = p_{r1,G} + 0.8(p_{r2,G} - p_{r3,G}) \tag{39}$$

where

$$\begin{aligned} v_{i,G+1} &= (\hat{a}_p, \hat{b}_p, \hat{c}_p, \hat{a}_y, \hat{b}_y, \hat{c}_y), \\ p_{s,G} &= (a_p, b_p, c_p, a_y, b_y, c_y), \\ s &= r1, r2, \;\; \text{or} \;\; r3, \;\; \text{and} \;\; r1 \ne r2 \ne r3 \ne i \end{aligned}$$

The trial vector $u_{i,G+1}$ is generated with $C_r = 0.9$:

$$j_{rand_i} = \text{int}(rand_i[0,1] \times 6) \tag{40}$$

$$a_{pi,G+1} = \begin{cases} \hat{a}_{pi,G+1} & \text{if } (rand[0,1] \le 0.9 \;\text{ or }\; j_{rand_i} = 1) \\ a_{pi,G} & \text{otherwise} \end{cases} \tag{41}$$

$$b_{pi,G+1} = \begin{cases} \hat{b}_{pi,G+1} & \text{if } (rand[0,1] \le 0.9 \;\text{ or }\; j_{rand_i} = 2) \\ b_{pi,G} & \text{otherwise} \end{cases} \tag{42}$$

$$c_{pi,G+1} = \begin{cases} \hat{c}_{pi,G+1} & \text{if } (rand[0,1] \le 0.9 \;\text{ or }\; j_{rand_i} = 3) \\ c_{pi,G} & \text{otherwise} \end{cases} \tag{43}$$

$$a_{yi,G+1} = \begin{cases} \hat{a}_{yi,G+1} & \text{if } (rand[0,1] \le 0.9 \;\text{ or }\; j_{rand_i} = 4) \\ a_{yi,G} & \text{otherwise} \end{cases} \tag{44}$$

$$b_{yi,G+1} = \begin{cases} \hat{b}_{yi,G+1} & \text{if } (rand[0,1] \le 0.9 \;\text{ or }\; j_{rand_i} = 5) \\ b_{yi,G} & \text{otherwise} \end{cases} \tag{45}$$

$$c_{yi,G+1} = \begin{cases} \hat{c}_{yi,G+1} & \text{if } (rand[0,1] \le 0.9 \;\text{ or }\; j_{rand_i} = 6) \\ c_{yi,G} & \text{otherwise} \end{cases} \tag{46}$$

$$u_{i,G+1} = (a_{pi,G+1}, b_{pi,G+1}, c_{pi,G+1}, a_{yi,G+1}, b_{yi,G+1}, c_{yi,G+1}) \tag{47}$$

If the trial vector produced violates any bound on the variables, it is fixed at the bound violated.

$$u_{j,i,G+1} = \left\{ \begin{array}{ll} +5 & \text{if } u_{j,i,G+1} \ge +5 \\ -5 & \text{if } u_{j,i,G+1} \le -5 \end{array} \right\}, \quad u_j = a_p, b_p, c_p, a_y, b_y \;\; \text{or} \;\; c_y$$

4.2.3. Selection

The values of J that the trial vector $u_{i,G+1}$ and the target vector $p_{i,G} = (a_{pi,G}, b_{pi,G}, c_{pi,G}, a_{yi,G}, b_{yi,G}, c_{yi,G})$ produce are computed. The one that produces lower J becomes $p_{i,G+1}$, that is, a part of the next generation, and the other is discarded. By using this kind of tournament ('knockout') selection between every pair of parent and child, DE ensures two things. One is that the diversity of the population is never lost, as may happen if the whole of the parent population is replaced by the child population, as in some evolutionary algorithms like the GA. The second is that, elitism, or the practice of preserving the best trial solution in a population is automatically ensured: if the best target vector is better than the competing trial vector, it gets passage to the next generation child population. If not, it is replaced by the new best (trial) vector in the whole population.

Mutation, crossover and selection are sequentially carried out for $G = 1, 2, \ldots G_{max}$ generations. The G_{max}th generation individual $(a_{pl,G_{max}}, b_{pl,G_{max}}, c_{pl,G_{max}}, a_{yl,G_{max}}, b_{yl,G_{max}}, c_{yl,G_{max}})$ that produces the least J is the best solution obtained from the procedure.

The cost or fitness function J is computed as follows: a trial solution $\bar{x}_{i,G} = (a_{pi,G}, b_{pi,G}, c_{pi,G}, a_{yi,G}, b_{yi,G}, c_{yi,G})$ is substituted in (20) and (21) to find f_p and f_y, and these are substituted in (18) and (19) to obtain the latax $\mathbf{A}_M(t)$ given by (17) for the time period $[t_0, t_f]$. For the constrained input case, the latax is fixed at $\pm\bar{a}_M$, for the time segments in which the latax exceeds the maximum bound $\pm\bar{a}_M$. This constrained latax is used to integrate (1) and (3) until the miss distance is less than the lethal radius assumed. The integration is done numerically using a fourth order Runge-Kutta method. J is calculated with the help of the t_f thus determined, and $A_m(t)$, using (36).

The application of DE to the missile guidance problem is shown in the form of a flowchart in Fig. 3. It is worth noting that this procedure is general enough to determine an optimal missile guidance law for any missile–target engagement problem, whose structure consists of a set of tunable parameters that have to be arrived at by trial and error.

4.3. Online-implementation

As cited in Section 3.1, the previous application of DE to optimal control [10] was offline, and took more than a hundred seconds to compute the control for a dynamic process that was of less than one second duration. The solution process for AAPN too, proposed by Sim et al. [39] was offline. Consequently, computation time was

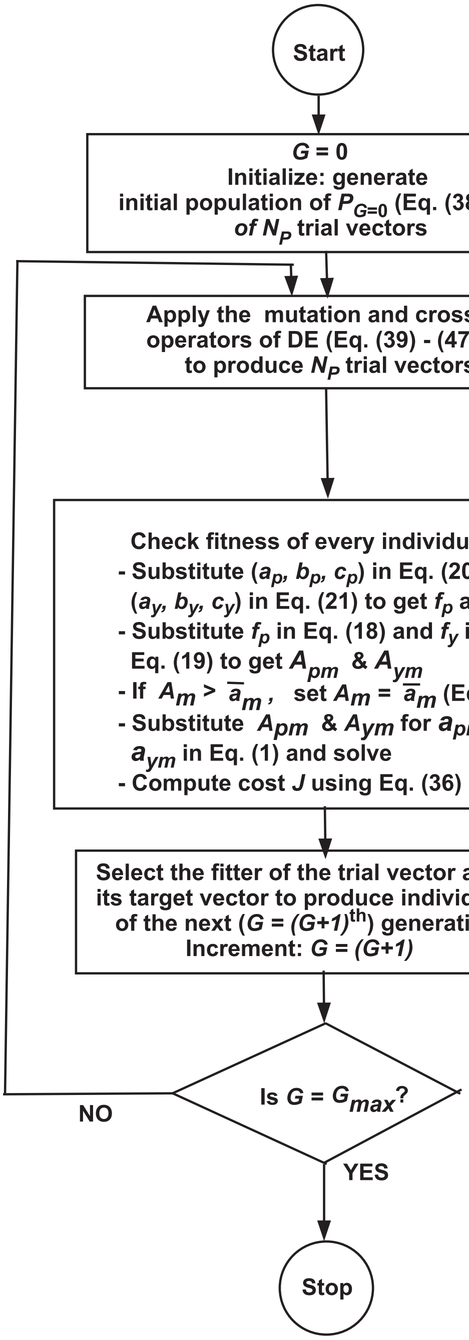

Start
$G = 0$
Initialize: generate initial population of $P_{G=0}$ (Eq. (38)) of N_p trial vectors
Apply the mutation and crossover operators of DE (Eq. (39) - (47)) to produce N_p trial vectors
Check fitness of every individual:
- Substitute (a_p, b_p, c_p) in Eq. (20) and (a_y, b_y, c_y) in Eq. (21) to get f_p and f_y
- Substitute f_p in Eq. (18) and f_y in Eq. (19) to get A_{pm} & A_{ym}
- If $A_m > \bar{a}_m$, set $A_m = \bar{a}_m$ (Eq. (37))
- Substitute A_{pm} & A_{ym} for a_{pm} & a_{ym} in Eq. (1) and solve
- Compute cost J using Eq. (36)
Select the fitter of the trial vector and its target vector to produce individual of the next ($G = (G+1)^{th}$) generation Increment: $G = (G+1)$
Is $G = G_{max}$?
NO
YES
Stop

Fig. 3. Application of DE to determine the DE-tuned-AAPN (DEPN-3D).

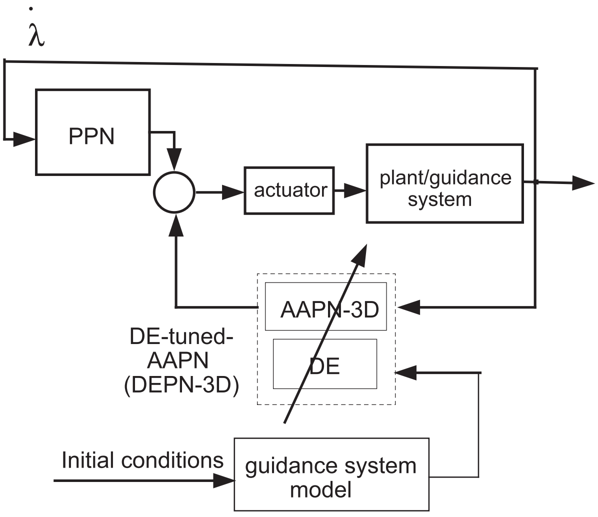


Fig. 4. Scheme for online implementation of DEPN-3D.

not a constraint in these references. As explained in Thangavelu and Pradeep [50] and Raghunathan and Ghose [31], for dynamic optimization or optimal control applications, EAs could be employed for finding solutions online, under any of the following circumstances: (i) the plant or system model is available, and the dynamics is slow enough to allow the computation time needed by the EA to arrive at the solution, (ii) a 'good enough' solution, not necessarily the best, is all that is needed for the application and (iii) massively parallel implementation is possible, so that the best solution is obtained in a small interval of time.

However, the missile guidance problem falls in none of the above three categories. The dynamics is very fast, and the best solution, the one that produces minimum J with acceptable miss distance, is the one that is acceptable. Though computing capacities of computers have been increasing rapidly, the onboard computing power on the missile is still not sufficient to allow massively parallel implementation. To overcome the above difficulties, this paper proposes a two loop hybrid control.

Fig. 4 shows the two loop hybrid control scheme for the online-implementable DE tuned AAPN (DEPN-3D) law, first proposed in Raghunathan and Ghose [31], and also used in the present work. The upper loop that includes the PPN would be in action until some intermediate, discrete time instants t_i, $i = 1, 2, \ldots, t_i < t_f$, where t_f is the flight time computed from the model used in the lower loop. Within this time instant, the DEPN-3D law would compute the best solution. In case no convergence is achieved within t_1 for some reason, the upper loop would continue to be applied until t_i, $i = 2, 3, \ldots$, at which convergence occurs. Thus, there is a backup guidance law in the upper loop, even if no convergence occurred for the entire duration of the engagement, thereby ensuring at least the primary requirement of capturability, even if the optimization fails and the secondary requirement of optimality cannot be achieved. This addresses the practical problem of failure of convergence of the optimizer, or algorithm adopted for optimization.

4.3.1. *Reducing the run time*

The basic methodology of online implementation is adapted from Raghunathan and Ghose [31]. This computationally challenging task of calculating the DEPN-3D guidance law in real-time is accomplished by the following steps:

(i) Increasing the number of coefficients in the polynomial functions f_p and f_y given by (20) and (21) does show a marginal improvement in the value of J. However, since f_p and f_y too need to be calculated for every individual in every generation, the number of coefficients in each of these equations is restricted to 3, as in (16). On the other hand, reducing the number of coefficients to 2 leads to results that either vary over too wide a range to be acceptable or takes too long to arrive at results that are within acceptable range.

(ii) Even after using only 3 coefficients in each polynomial function, there are six unknown coefficients a_p, b_p, c_p, a_y, b_y and c_y ($D = 6$) to be computed in the cost function evaluation. The number of unknowns is reduced to half by making $a_p = a_y = a$, $b_p = b_y = b$, and $c_p = c_y = c$, to reduce the computation time for online implementation.

This also reduces the population size, $N_P = 4D$ from 24 to 12.

(iii) Since it was found that there is reasonably good convergence within 20 generations of the DE, the DE simulation run is terminated after 20 generations. The best solution triple (a, b, c) obtained after 20 generations is used to calculate the DEPN-3D law. Keeping the total number of cost function evaluations for solving the problem $N_P \times G_{max}$ constant, one could also interchange the values used for N_P and G_{max}, without increasing the total computational effort. While this does produce the same result for this particular problem, it is conjectured that it is better to have the higher value for G_{max} than for N_P, since the possibility of premature convergence is better avoided by doing so, given that the topography of the solution space is unknown, and is different for different missile trajectories, due to differences in altitude, drag, and time duration of the trajectories.

(iv) The numerical solution of (1) and (3) that needs to be solved as a part of fitness evaluation of every solution vector in every generation is time consuming. To alleviate this, the step size in the RK-4 function used to solve (1) and (3) is increased to 0.1 s, and the miss distance for terminating the simulation is also increased to 25 m.

The coding and simulation is done in MATLAB® version 6.5.0 and SIMULINK® version 5 on a pentium IV, 3 GHz computer with 512 MB RAM and 4 GB virtual memory. To achieve fast execution, the integration of (1) and (3) is done in the SIMULINK environment, running in accelerator mode. Further, the in-built features of SIMULINK accelerator that enable even faster execution are made use of. The rest of the simulation that includes applying DE to the missile guidance problem is done in the MATLAB environment.

5. Implementation and simulation results

5.1. Realistic model of the missile

The basic dynamic model for the missile that includes aerodynamic effects is adapted from Cheng and Gupta [6], Kee, Dong and Siong [20], Ratnoo and Ghose [33,34].

The aerodynamic drag D that appears in (1) is modelled as $D = D_0 + D_i$, where the zero-lift drag $D_0 = C_{d0}Qs$, the induced drag $D_i = Km^2a_m^2/(Qs)$, the induced drag coefficient $K = 1/(\pi A_r e)$, the dynamic pressure $Q = \frac{1}{2}\rho V_m^2$, where s is the planform reference area, A_r the aspect ratio, e the efficiency factor, the atmospheric density, $\rho(h) = 1.5579 - 1.058 \times 10^{-4}h + 3.725 \times 10^{-9}h^2 - 6.0 \times 10^{-14}h^3$, where $h \in [0, 20,000]$ is the altitude.

The coefficient $C_{d0} = 0.02,\ M < 0.93\,;\, C_{d0} = 0.02 + 0.2(M - 0.93),\ 0.93 \le M < 1.03\,;\, C_{d0} = 0.04 + 0.06(M - 1.03),\ 1.03 \le M < 1.10\,;\, C_{d0} = 0.0442 + 0.007(M - 1.10),\quad M \ge 1.10$ where M is the Mach number, and the reference area s is 1 m^2. The computed values of K are: $K = 0.2,\ M < 1.15\,;\, K = 0.2 + 0.246(M - 1.15),\ M \ge 1.15$. The Mach number, $M = \frac{V_m}{\sqrt{1.4RT}}$, $R = 288$, where the temperature variation with altitude is given by $T = 288.16 - 0.0065\,h$ K, $h \le 11,000$ m ; $T = 216.66$ K, $h > 11,000$ m.

The thrust profile and the mass are assumed to be: $\Gamma = 12,000$ N, $m = 165 - 4t$ kg, for $0 \le t \le 10$ s ; $\Gamma = 2000$ N, $m = 125 - 1t$ kg, for $10 < t \le 40$ s ; $\Gamma = 0$ N, $m = 95$ kg, for 40 s $< t$.

5.2. Missile–target engagement variables

The proposed method is implemented on the following problem:

Without loss of generality, for simplicity and convenience of calculation, the initial LOS is assumed to be along the X_I axis ($\gamma_L = \phi_L = 0$). The other initial conditions of the engagement are: $x_m = y_m = y_t = 0$, $z_m = z_t = 3000$, $x_t = 3000$, $V_m = 400$, $V_t = 200$.

The time-weight $W_t = 10^4$, the value used in Raghunathan and Ghose [31] and Sim et al. [39]. Though practical values of the navigation constant N, is usually set at between 2 and 4 (p. 361, Lin [23]), this value is chosen to be 3, since this value has been proven to be optimal at least under some conditions [4]. The lethal radius of the missile is assumed to be 10 m in the simulations used for validation. The other nominal conditions assumed are a lag-free guidance system ($T = 0$), and a 15g constraint on latax ($\bar{a}_M = 147$ m/s^2). The interception or capture is deemed to be successful only if the miss distance falls to less than 10 m without the missile terminal speed ($V_m(t_f)$) falling to less than a cutoff speed of 350 m/s, and the integration of (1) and (3) is stopped at this time instant.

Following in the footsteps of Morgan et al. [27], Choi et al. [8] and Hossain et al. [15], the guidance law used for comparison against the DEPN-3D law of this paper is the PPN, which is a ready benchmark, and one that has been extensively studied in the literature and believed to have been widely implemented in practice. As the law superior to even the PPN when target maneuver is known [56], the augmented PPN (APPN) (covered in the Appendix for the sake of completeness) too has been used as another benchmark for comparison in this paper. The final benchmark used in this paper is the AAPN of Sim et al. [39], since it is the original law upon which the past [31] and present works of the authors are loosely based on.

Although the method proposed in this paper is online-implementable, the results of the offline-implementation of the proposed method are also shown, for a fair comparison with the AAPN, since the AAPN is an offline law.[2] This may also be useful in the scenario that, if the target variables are known prior to interception, the optimal control and trajectories may be calculated in advance.

The time required for computing the online version of the DEPN-3D law (the lower loop in Fig. 4) is found to be mostly just over 5 s, and always under 6 s, for $D = 3$, $N_P = 12$, $G_{max} = 20$ (the values arrived at in Section 4.3.1). Hence, the first intermediate time instant t_1 (discussed in Section 4.3) is fixed at 6 s. For the problem in this paper, since convergence always occurred within 6 s, there was no need to consider further intermediate time instants ($t_2, t_3, \ldots$).

5.3. Nominal case with a receding, non-maneuvering target (Case 1(a))

A non-maneuvering ($a_{yt} = a_{pt} = 0$), receding target is assumed to be flying away from the missile in an exact tail chase initially ($\gamma_t = \phi_t = 0°$), and the missile has very high initial heading errors of $h_p = -30°$, $h_y = 125°$.

The performance of the various guidance laws under nominal conditions are shown in Table 1 (Case 1(a)) and Fig. 6(a) and (b). The DEPN-3D law is labeled as DEPN in all the graphs, owing to space constraint. The performance index J of the PPN law is shown as the base value of 100% for easy comparison. It is seen from Table 1 that, both the online and offline implementations of the DEPN-3D perform better than the other laws, with respect to three of the five criteria used for comparison: t_f, J and $V_m(t_f)$. The performance of all laws as per the fourth criterion, the a_m(max) is the same: all laws demand the limiting value of the latax, $\bar{a}_m$. This is because of latax saturation which occurs since the heading error is very high. The fifth criterion of the limiting values of target maneuver for which capture takes place is inapplicable in the nominal case.

Comparing the online implementable laws, the online implementation of DEPN-3D is constrained from performing even better by the fact that it is applied only from $t_1 = 6$ s, and it is the PPN that is applied until then. In spite of this, it is a better performer than the PPN and AAPN. Fig. 6(a) shows how this is achieved. For the same latax saturation limit of 15g = 147 m/s^2 that is reached by all the laws, it applies more pitch latax from t_1 onwards, thus making the missile rise to a higher altitude in the atmosphere. This enables the missile to take advantage of the reduced drag at higher altitude. However, this has to be traded off against the continuous loss

[2] The classic laws, PPN and APPN are online implementable, as indicated in Tables 1 and 2.

Table 1
Case 1 – receding target, initial separation = 3000 m, $h_p = -30^\circ$, $h_y = 125^\circ$.

		PPN (online implementable)	AAPN (offline only)	DEPN-3D (offline implementable)	DEPN-3D (online implementable)	APPN (optimal law; online implementable)
	(a) nom: $a_T = 0$	18.96	18.19	16.63	17.86	18.96
	(b) $a_{yt0} = -1.52g$,	20.71	19.55	17.59	18.70	19.63
	$a_{pt0} = +1.52g$,∼(B-t-B, most unfavorable mnvr)					
	(c) $a_{yt0} = -1.52g$,	19.47	18.43	16.74	17.70	20.33
	$a_{pt0} = -1.52g$,∽(B-t-B, least unfavorable mnvr)					
t_f (s)	(d) $a_{yt} = -0.39g$,	21.45	20.32	18.21	19.91	20.97
	$a_{pt} = +0.39g$, step (unfavorable mnvr)					
	(e) $a_{yt} = -0.39g$,	18.04	17.46	16.59	17.40	18.25
	$a_{pt} = -0.39g$, step (favorable mnvr)					
	(a) nom: $a_T = 0$	100% = 2.387×10^5	97.65	92.0	95.98	100
	(b) ∼(B-t-B)	107.0	103.35	96.52	99.87	107.16
J (%)	(c) ∽(B-t-B)	103.9	100.80	94.01	96.98	104.52
	(d) ∓0.39g, step	104.78	101.93	96.48	100.67	104.32
	(e) −0.39g, step	96.69	94.55	91.41	93.76	97.07
a_m(max.), m/s^2		147 (=$\bar{a}_m$), for all cases, (a)–(e)				
	(a) nom: $a_T = 0$	381.64	397.44	446.15	407.78	381.64
$V_m(t_f)$	(b) ∼(B-t-B)	350.08	370.13	419.36	388.03	367.97
(m/s^2)	(c) ∽(B-t-B)	368.77	387.61	435.19	405.49	355.15
	(d) ∓0.39g, step	350.59	366.93	415.45	377.07	357.25
	(e) −0.39g, step	396.48	410.82	447.25	417.32	391.12
Capture	(b) ∼(B-t-B)	1.52g	2.52g	3.73g	3.93g	1.85g
upto	(c) ∽(B-t-B)	3.07g	3.92g	4.35g	4.62g	1.86g
a_{T_0}(max.)	(d) ∓, step	0.39g	0.59g	1.16g	0.72g	0.50g
	(e) -, step	2.09g	2.09g	2.15g	2.10g	1.87g

of kinetic energy due to drag on account of the longer flight duration of the longer path involved in doing so. Thus, the climb has to be only to an optimal height, as has been noted in the literature [6,53,32,40]. The DEPN-3D law accomplishes this automatically, given the model of the atmosphere, and the aerodynamic model of the missile. The offline DEPN-3D faces the least drag, and the PPN the most, as may be deduced from Fig. 6(b).

The offline implementation of DEPN-3D is the best performer of all laws, since the guidance algorithm can be applied from the beginning of the engagement ($t = 0$), and the tuning of the coefficients a, b, and c is much better than the default values of these coefficients in the AAPN law, which is also an offline implementation.

It is seen from Table 1 that the terminal speed $V_m(t_f)$ of the missile is inversely related to the flight time t_f, since the missile loses its speed at every instant of flight time due to aerodynamic drag. Due to the shorter flight times, the DEPN-3D laws are the best performers in this respect as well.

A complicating factor with the use of Euler angles in 3-D engagement scenarios is the singularity that arises for some values of the variables. This complication arises in the integration of (1) and (3), as for instance, when the angles γ_t and γ_m cross 90°, making $\dot{\phi}_t$ and $\dot{\phi}_m$ shoot to very high values at these points. This difficulty is tackled by limiting these values to |0.2| at these points.

The trajectories of the missile and the target have been included for better understanding and visualization of the performance of the guidance laws, for all the cases in this paper.

A typical convergence curve of the solution for the online-implementable DEPN-3D is shown in Fig. 5. The variation in the values of J obtained in repeated trials, after 20 generations, was less than 0.1 %, obviating the need for any detailed statistical analysis.

Fig. 6.

5.4. Receding, bank-to-bank maneuvering target (Cases 1(b) and (c))

The true test of any guidance law is when it faces off-nominal conditions. By far the most important off-nominal condition that causes performance degradation is the target maneuver. Unsurprisingly then, one of the first improvements to PN towards optimizing its performance was to address this issue by coming up with an augmentation term involving target maneuver, leading to the APPN law (see Appendix A).

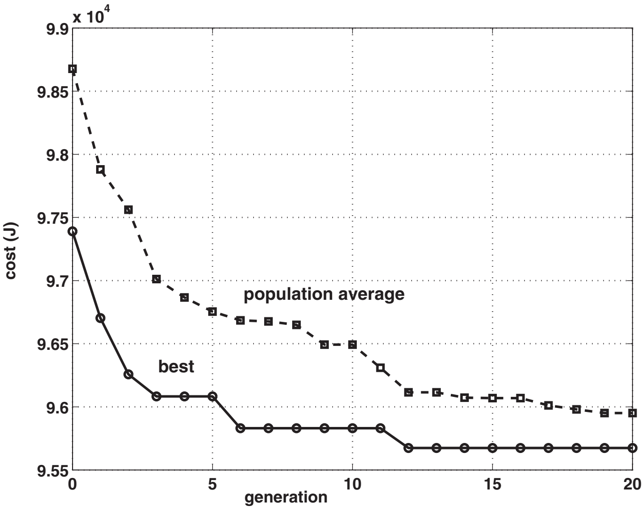
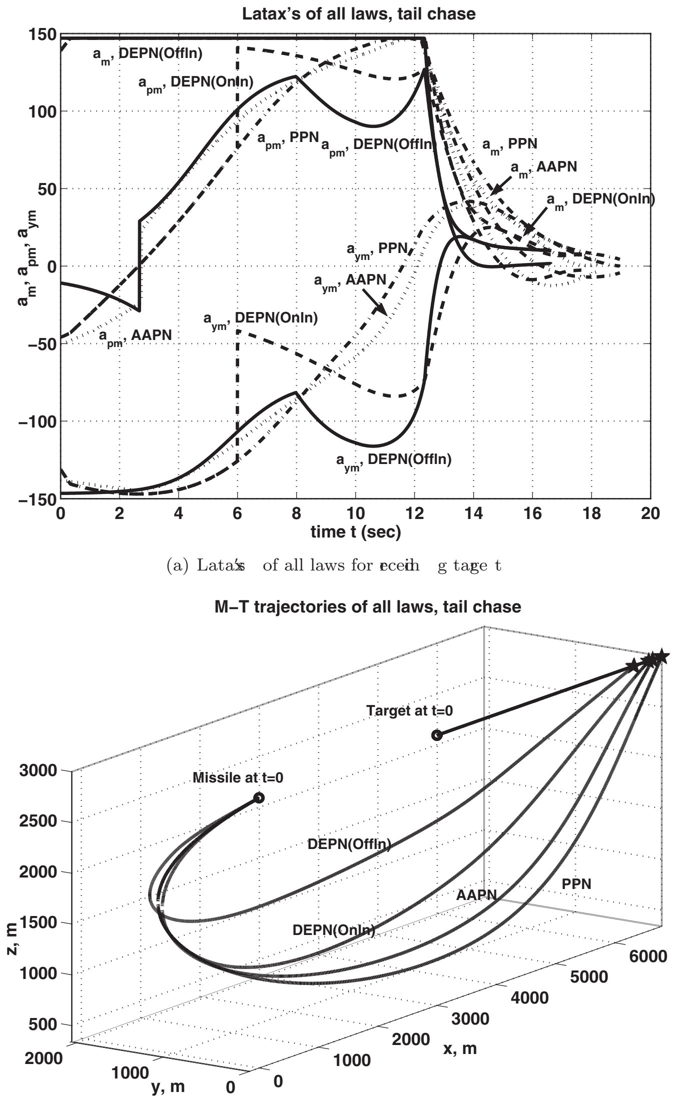


Fig. 5. Typical convergence of the solution.

The robustness of the guidance laws in this paper is tested by letting the target undertake unanticipated maneuvers. The target is assumed to be undertaking the two most common evasive maneuvers in practice: bank-to-bank (B-t-B), and step. Further, for ease of comparison, the target is assumed to be maneuvering equally in the pitch and the yaw planes.

The performance of all the laws to a B-t-B maneuvering target is considered first. The results are shown in Table 1 as Case 1(b). The maximum target maneuver magnitude for which all laws intercept the target is determined to be 1.52g. As can be expected, the performance of all the laws is poorer when compared with the nominal



(a) Latax's of all laws for receding target

(b) Missile-target trajectories of of all laws

Fig. 6. Case 1(a) – receding target, initial separation = 3000 m, $h_p = -30^\circ$, $h_y = 125^\circ$, $\bar{a}_M = 15g$.

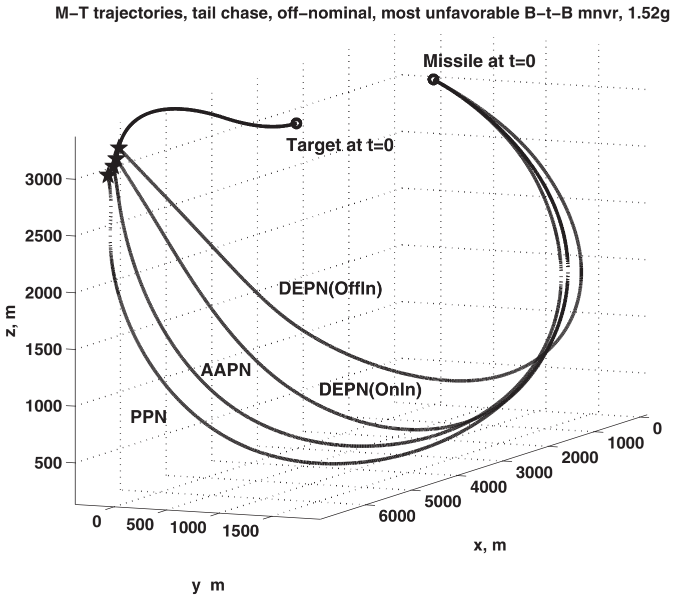


Fig. 7. Case 1(b) – M–T trajectories, tail chase, off-nominal, most unfavorable B-t-B mnvr, 1.52g.

case (Case 1(a)). However, the pattern of results for the nominal case is seen to be repeated in this case too. The least values of J is for the offline DEPN-3D, and the maximum value is for APPN, which are slightly more than that for PPN. This prima facie surprising result is readily explained by the fact that, armed with target maneuver information, it compensates the target maneuver from t_0 itself, a futile effort that is more costly in terms of J in the more practical world that the aerodynamic model represents – when the target is far away. The maximum value of $V_m(t_f)$ is for offline DEPN-3D, and the minimum value is for PPN. This is explained by the fact that the DEPN-3D has the least flight time, and hence suffers the least energy loss due to drag. Similar reasoning applies to PPN.

The maximum magnitude of the unanticipated target maneuver for which the target can be intercepted is the minimum for PPN (1.52g), and more than double this value for online DEPN-3D (3.93g). Interestingly, the corresponding value for the offline DEPN-3D law (3.73g) is lower than for the online DEPN-3D law. This is explainable by the fact that, for a sinusoidal maneuvering target, a guidance law that applies higher latax from t_0 itself is applying too much latax too soon, expends energy in vain, and finds itself weaker in the later stages of the engagement. The missile–target trajectories for this case are shown in Fig. 7.

Through simulations, the B-t-B maneuver that is least unfavorable to interception is found to be: both a_{yt} and a_{pt} negative for the first 6 s, of the opposite sign for the next 6 s, and so on. This case is shown as Case 1(c) in Table 1. For comparison with the most unfavorable case (Case 1(b)), the magnitude of the maneuver is restricted to the same magnitude of 1.52g in each plane. The performance of all the laws is expectedly poorer than for the nominal case (Case 1(a)), but better than that for the most unfavorable case. The pattern of the best results for offline DEPN-3D and poorest results for APPN is again seen to be repeated in this case too. By compensating for a downward moving target from t_0 itself, the APPN law loses altitude and hence requires even more time and energy in the later stages of the engagement than PPN for interception. This is reflected in the values of t_f and J. Since this case is less unfavorable than Case 1(b), all laws except the APPN can intercept with higher target maneuver levels than for Case 1(b). For instance, the PPN can intercept up to 3.07g for this case, as against 1.52g for Case 1(b), and the offline DEPN-3D can intercept up to 4.62g for this case, as against 3.93g for Case 1(b). The corresponding figures for most unfavorable maneuver and least unfavorable maneuver are 3.73g and 4.35g for the online DEPN-3D. These figures suggest that, when faced with an unanticipated B-t-B target maneuver, the PPN is more sensitive to the direction and magnitude of the maneuvers than the DEPN-3D, and having target maneuver information that is compensated from t_0 itself like the APPN does, is either no advantage at all (Case1(b)), or a disadvantage (Case1(c), in a realistic scenario. The AAPN is seen to be less sensitive than the PPN, and more sensitive than DEPN-3D, as per this criterion. With its performance against PPN itself not being so favorable, the APPN is definitely a poor performer against the DEPN-3D laws, as per four out of the five criteria chosen for comparison, the exception being the maximum latax that is same for all laws.

The logical conclusion is that, in a realistic scenario, a more intelligent law like the DEPN-3D law is needed for optimal performance, rather than a basic law like the APPN that merely nullifies target maneuver without any means of knowing whether doing so is beneficial or not.

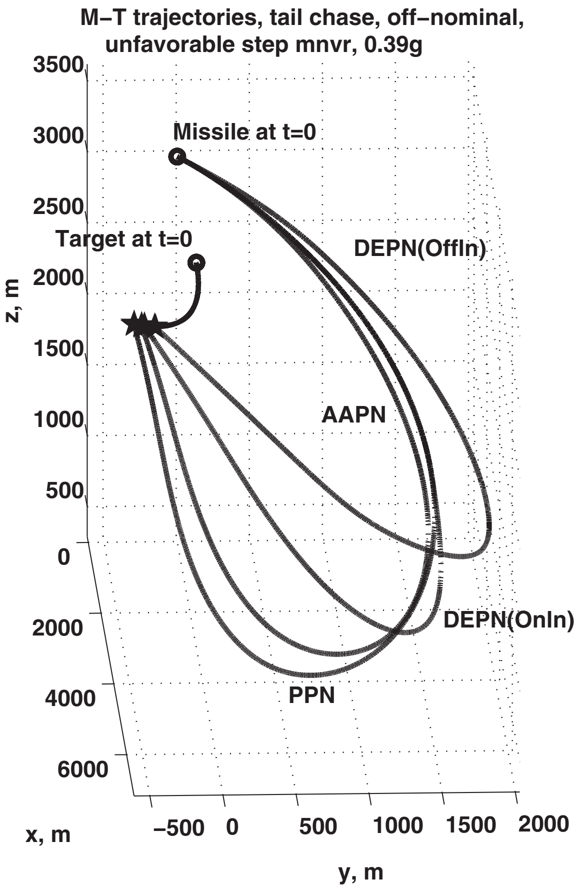
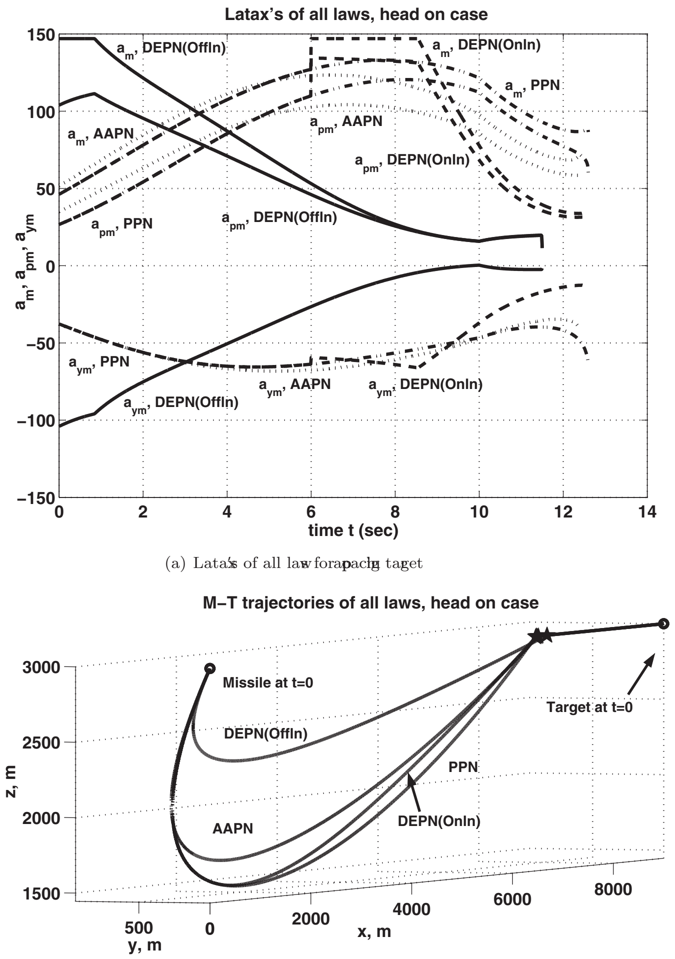


Fig. 8. Case 1(d) - M–T trajectories, tail chase, off-nominal, unfavorable step mnvr, 0.39*g*.

5.5. Receding, step maneuvering target (Cases 1(d) and (e))

The performance of the laws for a step maneuvering target is studied next. By trying out all combinations as for the B-t-B maneuver cases, the most unfavorable step maneuver for the PPN law is determined to be a negative value of a_{yt}, and a positive value for a_{pt}.The maximum magnitude for which all laws intercept for this case (Case 1(d) in Table 1) is found to be 0.39*g*, which is just more than a fourth of the 1.52*g*, the magnitude in the most unfavorable B-t-B case (Case 1(b)). This would suggest that the step maneuver is more effective than a B-t-B maneuver for the target. Examining the case in which the step maneuver actually favors interception by the PPN (Case 1(e)), the maximum maneuver magnitudes that allow interception are less than those for Case 1(b) or Case 1(c). This suggests that, at least for the high heading error scenarios considered in this paper, the step maneuver is more effective than the B-t-B maneuver from the target's point of view. Conversely, from the missile point of view, all the guidance laws including the DEPN-3D law are more robust to B-t-B maneuver than the step maneuver. The optimal APPN law performs marginally better than PPN law for Case1(d), since the compensation term in the latax includes an upward pitching component that proves beneficial later on. But in Case1(e), this compensation for downward maneuver makes the missile lose height initially, resulting in a performance that is poorer than even the PPN as per four out of the five criteria for comparison. The best two performers are the offline DEPN-3D and the online DEPN-3D for the step maneuver too. The missile–target trajectories for this case are shown in Fig. 8.



(a) Latax's of all laws for approaching target

(b) Missile-target trajectories of all laws

Fig. 9. Case 2(a) – approaching target, initial separation = 9000 m, $h_p = -45^\circ$, $h_y = 45^\circ$, $\bar{a}_M = 15g$.

5.6. Nominal case with an approaching target (Case 2(a))

The scenario of an approaching target, that arises more often in tactical missile guidance than that of a receding target, is considered next. The optimal solution for the AAPN of Sim et al. [39] has been obtained for the receding target, but it is still a better performer than the PPN even for the approaching target. Hence it is retained in this paper for comparison, as a useful benchmark. The nominal case of a non-maneuvering ($a_{yt} = a_{pt} = 0$), approaching target flying exactly head-on to the missile initially ($\gamma_t = 0^\circ$, $\phi_t = 180^\circ$) is considered, as in Raghunathan and Ghose [31]. The initial position of the missile is same as in the beginning of Section 5.2. The only change is the initial separation x_t, which is assumed to be 9000 m, instead of 3 000 m. The initial heading errors are assumed to be $h_p = -45^\circ$, $h_y = 45^\circ$.

The performance of all the laws are shown in Table 2 and Fig. 9(a) and (b). In this case, the online DEPN-3D is a poorer performer than the AAPN by all the criteria, since it uses a slightly lower latax than the AAPN until t_1 (Fig. 9(a)), and the flight time remaining after t_1 is less, as compared to Cases 1(a)–(e). This does not allow it the scope to improve further. This is also seen in Fig. 9(b); the online DEPN-3D trajectory dips as low as that of the PPN, before rising somewhat to improve on the PPN. Between the two offline methods, the offline DEPN-3D is expectedly a much superior performer by four of the five criteria (t_f, J, $V_m(t_f)$ and the max. a_{T0} up to which target capture occurs), since its design uses the same initial engagement geometry with the same realistic engagement model with which it is

Table 2
Case 2 - approaching target, initial separation = 9000 m, $h_p = -45^\circ$, $h_y = 45^\circ$.

		PPN (online implementable)	AAPN (offline only)	DEPN-3D (offline implementable)	DEPN-3D (online implementable)	APPN (optimal law; online implementable)
t_f (s)	(a) nom: $a_T = 0$	12.60	12.37	11.49	12.53	12.60
	(b) $a_{yt0} = -2.86$ g, $a_{pt0} = -2.86$ g,∽(B-t-B, most unfavorable mnvr)	12.84	12.61	11.76	12.66	NI[a]
	(c) $a_{yt0} = +2.86$ g, $a_{pt0} = +2.86$ g,~(B-t-B, less unfavorable mnvr)	13.29	13.03	12.07	13.26	12.56
	(d) $a_{yt} = -9.16$ g, $a_{pt} = +9.16$ g, step (unfavorable mnvr)	17.36	17.19	16.65	17.38	19.61
	(e) $a_{yt} = -9.16$ g, $a_{pt} = -9.16$ g, step (favorable mnvr)	20.48	20.01	17.66	19.78	NI[a]
J (%)	(a) nom: $a_T = 0$	100% =1.363 × 10^5	91.64	69.80	94.79	100
	(b) −2.86 ∽(B-t-B)	104.48	96.11	67.59	90.02	NI[a]
	(c) +2.86 ~(B-t-B)	102.27	95.89	81.07	102.13	89.80
	(d) ∓9.16g, step	110.42	110.71	111.37	114.31	154.20
	(e) −9.16 g, step	125.09	122.01	104.62	121.28	NI[a]
a_m(max.) m/s^2	(a) nom: $a_T = 0$	133.06	123.40	147(= $\bar{a}_m$)		133.06
	(b) −2.86 ∽(B-t-B)	147(= $\bar{a}_m$)			141.37	NI[a]
	(c) +2.86 ~(B-t-B)	144.53	141.09	147(= $\bar{a}_m$)		127.15
	(d) ∓9.16g, step	147(= $\bar{a}_m$)				
	(e) −9.16 g, step	147(= $\bar{a}_m$)			117.77	NI[a]
$V_m(t_f)$ (m/s^2)	(a) nom: $a_T = 0$	586.57	604.98	672.68	595.32	586.57
	(b) −2.86 ∽(B-t-B)	562.33	580.09	650.45	581.93	NI[a]
	(c) +2.86 ~(B-t-B)	570.17	588.51	653.0	574.12	611.37
	(d) ∓9.16g, step	471.08	484.81	527.41	474.02	451.84
	(e) −9.16 g, step	354.32	366.70	435.26	375.14	NI[a]
Capture upto a_{T_0}(max.)	(b) ∽(B-t-B)	2.86g	4.10g	20.42g	15.99g	1.17g
	(c) ~(B-t-B)	9.30g	9.45g	19.93g	20.21g	10.66g
	(d) ∓, step	9.16g	9.77g	21.98g	19.10g	12.85g
	(e) −, step	20.37g	20.66g	22.73g	20.13g	4.09g

[a] NI = no interception.

tested. Whereas, the AAPN was designed for tail chase condition, for a kinematic model. This superior performance is due to the much higher latax (equal to the saturation limit $\bar{a}_m$) employed by the offline DEPN-3D from initial time t_0 itself (Fig 9(a)).

5.7. Approaching, bank-to-bank maneuvering target (Cases 2(b) and (c))

As with Case 1(b), the most unfavorable B-t-B target maneuver for the PPN law is determined by trial and error. For this case (Case 2(b)), the optimal APPN does not even intercept the target since, in a limiting case, it nullifies an initially beneficial target maneuver that could actually facilitate interception, disadvantaging itself by losing height in the process of doing so. This is reversed in Case 2(c), in which the missile guided by APPN gains height initially, and hence, achieves interception. For the remaining three laws, the pattern of results for the nominal case of 2(a) is repeated, with the performance ranking between these being offline DEPN-3D, AAPN, and online DEPN-3D, in descending order.

5.8. Approaching, step maneuvering target (Cases 2(d) and (e))

The same pattern of results is repeated for step target maneuver, with the APPN in Case 2(e) being similar to Case 2(b) in failing to achieve interception, The logical explanations of why these happen are also similar Case 1(e): faced with an initial target maneuver that favors interception, the APPN nullifies this, since it has no means of knowing that the maneuver is actually a beneficial one. It expends extra energy in the process than the PPN, and hence is not able accomplish what the PPN does, for what are limiting values of target maneuvers for the PPN. Case 2(d) is somewhat like Case 2(c), in being able to achieve interception: the latax that includes the compensation term for enables it to rise in altitude initially. Between the remaining three laws, the offline DEPN-3D law is again the best performer like in the rest of the cases considered, followed by AAPN, and online DEPN-3D.

In conclusion, the APPN does not have any edge over the PPN by virtue of its having the extra information of target maneuver. This edge would be available only if even more extra information of whether the maneuver is beneficial or otherwise is available for use. The DEPN-3D laws have this edge, and are naturally the best performers of the lot.

6. Conclusions

A completely new approach to optimal missile guidance, based on a 3-D realistic dynamic model, and using an evolutionary algorithm that is online implementable has been presented. By and large, the DEPN-3D laws designed by using this approach are shown to be best performers, and the most robust to target maneuver, in the overall analysis of all the laws studied in this work.

A comparison of the proposed laws with the optimal APPN reveals that, in a realistic scenario, a more intelligent law like the DEPN-3D law is needed for optimal performance, rather than a basic law like the APPN that merely nullifies target maneuver with no means of knowing whether doing so is beneficial or not.

Unlike the closed form optimal guidance laws that require time-to-go information, the DEPN-3D laws presented here does not require time-to-go information for its implementation.

A detailed study of the effects of the variation in the tuning parameters of DE may be conducted as future investigation, for issues like speed of convergence and faster implementation.

Given the fact that no optimization algorithm can work equally well on all classes of problems, it is felt that there exists considerable scope for studying whether there can be a better match between the optimization problem in the paper and the DE variant used to solve it. The problem can be studied for specific features that can be exploited for faster optimization by the state-of-the-art DEs that have appeared in the literature. Expanding the scope further, would be interesting to compare the performance of other metaheuristic methods against the results presented in this paper.

Acknowledgements

The authors would like to thank their anonymous reviewers for their valuable remarks.

Appendix A.

The Augmented PPN (APPN)

The PPN has been proven to be the energy-optimal guidance law at least under some conditions, for a non-maneuvering target. For a maneuvering target, this optimality of the PPN can be preserved by adding an augmentation term that compensates for the target maneuver. The compensation term includes the target latax normal to the LOS, and this term nullifies the rate of rotation of LOS due to the target maneuver. This law, known as the augmented PPN (APPN), is given by

$$a_{yt,\,\mathrm{APPN}} = a_{yt,\,\mathrm{PPN}} + \frac{1}{2}Na_{yt}\cos\psi_t \tag{48}$$

$$a_{pt,\,\mathrm{APPN}} = a_{pt,\,\mathrm{PPN}} + \frac{1}{2}Na_{pt}\cos\theta_t \tag{49}$$

where the angles ψ_t and θ_t are as given in Fig. 1.

References

[1] F.P. Adler, Missile guidance by three-dimensional proportional navigation, Journal of Applied Physics 27 (1956) 500–507.

[2] S. Bezick, I. Rusnak, W.S. Gray, Guidance of homing missiles via nonlinear geometric control methods, Journal of Guidance, Control and Dynamics 18 (1995) 441–448.

[3] J. Brest, S. Greiner, B. Boskovic, M. Mernik, Z. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Transactions on Evolutionary Computation 10 (2006) 646–657.

[4] A.E. Bryson, Y.C. Ho, Applied Optimal Control, Hemisphere Publishing Corp., Washington, DC, 1975.

[4] U.K.E. Chakraborthy, Advances in Differential Evolution Volume 143 of Studies in Computational Intelligence, Springer-Verlag, Berlin, Heidelberg, 2008.

[6] V.H.L. Cheng, N.K. Gupta, Advanced midcourse guidance for air-to-air missiles, Journal of Guidance 9 (1986) 135–142.

[7] J.-P. Chiou, F.-S. Wang, A hybrid method of differential evolution with application to optimal control problems of a bioprocess system, in: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence, 1998, pp. 627–632.

[8] H.-L. Choi, M.-J. Tahk, H. Bang, Neural network guidance based on pursuit-evasion games with enhanced performance, Control Engineering Practice 14 (2006) 735–742.

[9] J.R. Cloutier, D.T. Stansbery, All-aspect acceleration-limited homing guidance, in: AIAA Guidance, Navigation and Control Conference, 1999, pp. 633–639.

[10] I.L.L. Cruz, L.G.V. Willigenburg, G.V. Straten, Efficient differential evolution algorithms for multimodal optimal control problems, Applied Soft Computing 3 (2003) 97–122.

[11] S. Das, A. Abraham, U.K. Chakraborty, A. Konar, Differential evolution using a neighbourhood-based mutation operator, IEEE Transactions on Evolutionary Computation 13 (2009) 526–553.

[12] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Transactions on Evolutionary Computation 15 (2011) 4–31.

[13] M. Guelman, A qualitative study of proportional navigation, IEEE Transactions on Aerospace and Electronic Systems 7 (1971) 364–371.

[14] M. Guelman, Proportional navigation with a maneuvering target, IEEE Transactions on Aerospace and Electronic Systems 8 (1972) 637–643.

[15] M. Hossain, A. Madkour, K. Dahal, L. Zhang, A real-time dynamic optimal guidance scheme using a general regression neural network, Engineering Applications of Artificial Intelligence 26 (2013) 1230–1236.

[16] F. Imado, T. Kuroda, S. Miwa, Optimal midcourse guidance for medium-range air-to-air missiles, Journal of Guidance 13 (1990) 603–608.

[17] M. Innocenti, Nonlinear guidance techniques for agile missiles, Control Engineering Practice 9 (2001) 1134–1144.

[18] M. Innocenti, F. Pellegrini, F. Nasuti, A VSS guidance law for agile missiles, in: AIAA Guidance, Navigation and Control Conference and Exhibit, 1997, pp. 179–188.

[19] S. Kannan, S.M.R. Slochanal, N.P. Padhy, Application and comparison of metaheuristic techniques to generation expansion planning problem, IEEE Transactions on Power Systems 20 (2005) 466–475.

[20] E.K. Kee, L. Dong, C.J. Siong, Near optimal midcourse guidance law for flight vehicle, in: AIAA Guidance, Navigation and Control Conference, 1998.

[21] K. Krishnakumar, D.E. Goldberg, Control system optimization using genetic algorithms, Journal of Guidance, Control and Dynamics 15 (1992) 735–739.

[22] G. Leng, Guidance algorithm design: a nonlinear inverse approach, Journal of Guidance, Control and Dynamics 21 (1998) 742–746.

[23] C.F. Lin, Modern Navigation Guidance and Control Processing, Prentice Hall, Inc., New Jersey, 1991.

[24] E. Mezura-Montes, J. Velazquez-Reyes, C.A.C. Coello, A comparative study of differential evolution variants for global optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO2006, Seattle, Washington, USA, 2006.

[25] Z. Michalewicz, C.Z. Janikow, J.B. Krawczyk, A modified genetic algorithm for optimal control problems, Computers & Mathematics with Application 23 (1992) 83–84.

[26] Z. Michalewicz, J.B. Krawczyk, M. Kazemi, C.Z. Janikow, Genetic algorithms and optimal control problems, in: Proceedings of the 29th Conference on Decision and Control, Honolulu, HI, USA, 1990, pp. 1664–1666.

[27] W. Morgan, H. Tharp, T. Vincent, Minimum energy guidance for aerodynamically controlled missiles, IEEE Transactions on Automatic Control 56 (2011) 2026–2037.

[28] D.S. Naidu, Optimal Control Systems, CRC Press LLC, Boca Raton, FL, 2003.

[29] K.V. Price, New Ideas in Optimization. Chapter An Introduction to Differential Evolution, McGraw-Hill International (UK) Limited, Berkshire, England, 1999.

[30] K.V. Price, R. Storn, Differential evolution – a simple evolution strategy for fast optimization, Dr. Dobb's Journal 22 (1997) 18–24.

[31] T. Raghunathan, D. Ghose, An online-implementable differential evolution tuned all-aspect guidance law, Control Engineering Practice 18 (2010) 1197–1210.

[32] P.A. Raju, D. Ghose, Empirical virtual sliding target guidance law design: an aerodynamic approach, IEEE Transactions on Aerospace and Electronic Systems 39 (2003) 1179–1190.

[33] A. Ratnoo, D. Ghose, SDRE based guidance law for impact angle constrained trajectories, in: AIAA Guidance, Navigation and Control Conference, 2007.

[34] A. Ratnoo, D. Ghose, State-dependent-riccati-equation-based guidance law for impact angle constrained trajectories, Journal of Guidance, Control and Dynamics 32 (2009) 320–325.

[35] C.-K. Ryoo, H. Cho, M.-J. Tahk, Optimal guidance laws with terminal impact angle constraint, Journal of Guidance, Control and Dynamics 28 (2005) 724–732.

[36] C.-K. Ryoo, H. Cho, M.-J. Tahk, Time-to-go weighted optimal guidance with impact angle constraints, IEEE Transactions on Control Systems Technology 14 (2006) 483–492.

[37] H. Seywald, R.R. Kumar, S.M. Deshpande, Genetic algorithm approach for optimal control problems with linearly appearing controls, Journal of Guidance, Control and Dynamics 18 (1995) 177–182.

[38] N.A. Shneydor, Missile Guidance and Pursuit, Horwood Publishing Limited, Chichester, West Sussex, England, 1998.

[39] Y.C. Sim, S.B. Leng, V. Subramaniam, An all-aspect near-optimal guidance law, Dynamics and Control 10 (2000) 165–177.

[40] A. Singh, D. Ghose, A.K. Sarkar, Launch envelope optimization of virtual sliding target guidance scheme, IEEE Transactions on Aerospace and Electronic Systems 45 (2009) 899–918.

[41] S.-H. Song, I.-J. Ha, A Lyapunov-like approach to performance analysis of 3-dimensional pure PNG laws, IEEE Transactions on Aerospace and Electronic Systems 30 (1994) 238–248.

[42] R. Storn, On the usage of differential evolution for function optimization, in: Proceedings of the 1996 Biennial Conference of the North American Fuzzy Information Processing Society – NAFIPS, 1996, pp. 519–523.

[43] R. Storn, Differential Evolution Home Page, 2002, Available at: http://www.icsi.berkeley.edu/~storn/code.html.

[44] R. Storn, Differential Evolution Research – Trends and Open Questions of Studies in Computational Intelligence, vol. 143, Berlin, Heidelberg, Springer-Verlag, 2008.

[45] R. Storn, K.V. Price, Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. Technical Report TR-95-012 Berkeley, 1995, Available at: ftp://ftp.icsi.Berkeley.edu/pub/techreports/1995/tr-95-012.ps.Z.

[46] R. Storn, K.V. Price, Minimizing the real functions of the ICEC 1996 contest by differential evolution, in: Proceedings of the 1996 IEEE Conference on Evolutionary Computation, 1996, pp. 842–844.

[47] R. Storn, K.V. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.

[48] M.-J. Tahk, C.-K. Ryoo, H. Cho, Recursive time-to-go estimation for homing guidance missiles, IEEE Transactions on Aerospace and Electronic Systems 38 (2002) 13–24.

[49] D. Taur, Nonlinear guidance and navigation of a tactical missile with high heading error., in: AIAA Guidance, Navigation and Control Conference and Exhibit, 2002, pp. 1–19.

[50] R. Thangavelu, S. Pradeep, An online implementable differential evolution tuned optimal guidance law, in: Genetic and Evolutionary Computation Conference, GECCO 2007, 2007, pp. 2190–2194.

[51] J. Vesterstrom, R. Thomsen, A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, in: Proceedings of the Congress on Evolutionary Computation, 2004, pp. 1980–1987.

[52] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1997) 67–82.

[53] S.-M. Yang, Analysis of optimal midcourse guidance law, IEEE Transactions on Aerospace and Electronic Systems 32 (1996) 419–425.

[54] P.-J. Yuan, M.-G. Chen, J.-S. Chern, Extended proportional navigation, in: AIAA Guidance, Navigation and Control Conference and Exhibit, 2000, pp. 1–10.

[55] D. Zaharie, Critical values for the control parameters of differential evolution algorithms, in: Proceedings of the 8th Mendel Conference Soft Computing, 2002, pp. 62–67.

[56] P. Zarchan, Tactical and Strategic Missile Guidance Volume 199 of Progress in Astronautics and Aeronautics, 4th ed., AIAA, Reston, VA, 2002.

[57] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Transactions on Evolutionary Computation 13 (2009) 945–958.